REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	5 March 1997	

4. TITLE AND SUBTITLE

Distributed ATM Network Command & Control Reference Architecture

5. FUNDING NUMBERS

N00024-96-C-4188

6. AUTHOR(S)

Wei-Cheng Robert Tung

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Multi-Media Communications, Inc.

6610 Rockledge Dr., Suite 168

Bethesda, MD 20817

8. PERFORMING ORGANIZATION REPORT NUMBER

--

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Navy
Naval Sea Systems Command
NAVSEA 03K11
Arlington, VA 22242-5160

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

N96-073

19980423 055

11. SUPPLEMENTARY NOTES

one

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

ATM is a commercial switching and transport technology that uses a distributed control capability for wide-area networks. The commercial ATM switches support distributed routing and distributed bandwidth-on-demand during the call set-up process. However the limitations of static QoS and bandwidth allocation provisioning plus the blocking characteristics of call-admission-control do not provide enough flexibility and network availability to support avy's mission critical applications in a cost-effective manner. To meet the avy's requirements, the following network control capabilities are needed:

- priority admission
- dynamic QoS adjustment
- dynamic bandwidth adjustment
- automatic service restoration

A test bed was developed to demonstrate selected key features of dynamic bandwidth management (through a table decision process) and network restoration (through the update of VPI/VCI tables).

14. SUBJECT TERMS

ATM, bandwidth-on-demand, dynamic QoS

15. NUMBER OF PAGES 11

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	unlimited

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

FINAL TECHNICAL REPORT
TOPIC
N96-073

FOR

NAVAL SEA SYSTEMS COMMAND

NAVSEA 03K11
ARLINGTON, VA 22242-5160

WEI-CHENG ROBERT TUNG

DISTRIBUTED ATM NETWORK COMMAND & CONTROL REFERENCE ARCHITECTURE

MARCH 5, 1997
CONTRACT No. N00024-96-C-4188

$69,840.00 SBIR Phase I

Prepared by
Multi-Media Communications, Inc.
6610 Rockledge Drive, Suite 168
Bethesda, MD 20817

1. Introduction

ATM is a commercial switching and transport technology that enables distributed control capability for wide-area networks in a flexible and dynamic manner. Current commercially available ATM technologies, such as enterprise ATM switches with Private Network-to-Network Interface (PNNI) [1], may support bandwidth-on-demand and routing in a distributed manner during the call set-up process. However, the limitations of static QoS and bandwidth allocation provisioning and the blocking characteristics of call admission control (CAC) of existing ATM networks may potentially reduce flexibility and availability of supporting Navy's mission critical applications, whenever and wherever needed, in a cost-effective manner. In addition, the current PNNI only supports network re-routing for new connections only (i.e., during the initial call set-up process). It can't not be used to restore existing connections when the network components fail. Thus, enhancement on existing ATM technology and standards are needed in order to provide priority admission with the best available QoS and survivability to support Navy's military applications whenever (normal and failure conditions) and wherever needed.

To meet the foregoing Navy's mission critical operations requirements, the following advanced network control capabilities are needed:

- Priority admission;
- Dynamic QoS adjustment;
- Dynamic bandwidth adjustment; and
- Automatic service restoration.

Among these network control capabilities, dynamic bandwidth management capability is the core feature that would allow the command and control system to support dynamic QoS adjustment, and automatic service restoration. This capability provides bandwidth increment during the active connection session, if needed, when the network resources are available for the requested adjustment. The dynamic QoS adjustment capability will allow mission critical connections access the network whenever and wherever needed. Automatic service restoration would allow Navy's mission critical operations intact when the ship is in the hostile war-fighting environment.

The goal of Phase I work is to study a distributed ATM-based command and control system reference architecture that would allow Navy to support mission critical operations whenever and wherever need, and to develop a testbed to demonstrate feasibility of selected key features of dynamic bandwidth management and service restoration. This architecture will be based on the novel application of proven real-time control methodology, applied by DARPA, Navy and private sectors on projects as diverse as high-speed signaling transport, adaptive QoS control and management, and Web-based signaling and control. The feasibility demonstration of these two core network control capabilities would help Phase II work, which develops the server-based command and control system for priority admission, dynamic QoS management and network self-

healing, to identify potential problems and technical bottlenecks of server-network interfaces and network reconfiguration management and control.

2. Adaptive ATM QoS-based Distributed Command and Control System Architecture

Commercially available ATM PNNI-based systems [1] may be used as a platform to design the considered adaptive QoS-based command and control system. This platform would provide signaling capability on demand and support QoS and priority on a static and distributed manner. The considered adaptive QoS command and control system can be implemented on top of an ATM-based command and control system in a centralized or distributed manner, depending on the network size, reliability and performance requirements. The centralized, adaptive QoS command and control system may be more appropriate for small and intermediate networks, while the distributed adaptive QoS server system may be designed for large scale networks. For the centralized QoS server system, the command and control function is executed at a central controller. By contrast, the command and control function in a distributed adaptive QoS system is distributed within the network and executed either at each ATM switch or a set of interacting smaller QoS servers.

Figure 1 depicts an advanced adaptive QoS-based command and control system architecture. This architecture includes a distributed ATM signaling and control transport network and a dynamic QoS server system that implements new system requirements as described in Section 1. The ATM switch, which provides the core transport capability for distributed command and control functions, may be implemented by an ATM switch having PNNI capability. In the proposed architecture, the QoS server is de-coupled from ATM switches to preserve the switch performance efficiency. In addition, the de-couple of ATM switches from the QoS server function allows this advanced command and control capability be added without changing the existing command and control system infrastructure. To increase system reliability, a backup dynamic QoS server may be deployed, and the secondary QoS server should operate in a synchronous mode with the primary QoS server. Furthermore, to reduce the probability of system congestion, load balancing can be performed between two QoS servers. Several distributed control-based server system designs can be found in Ref. [2].

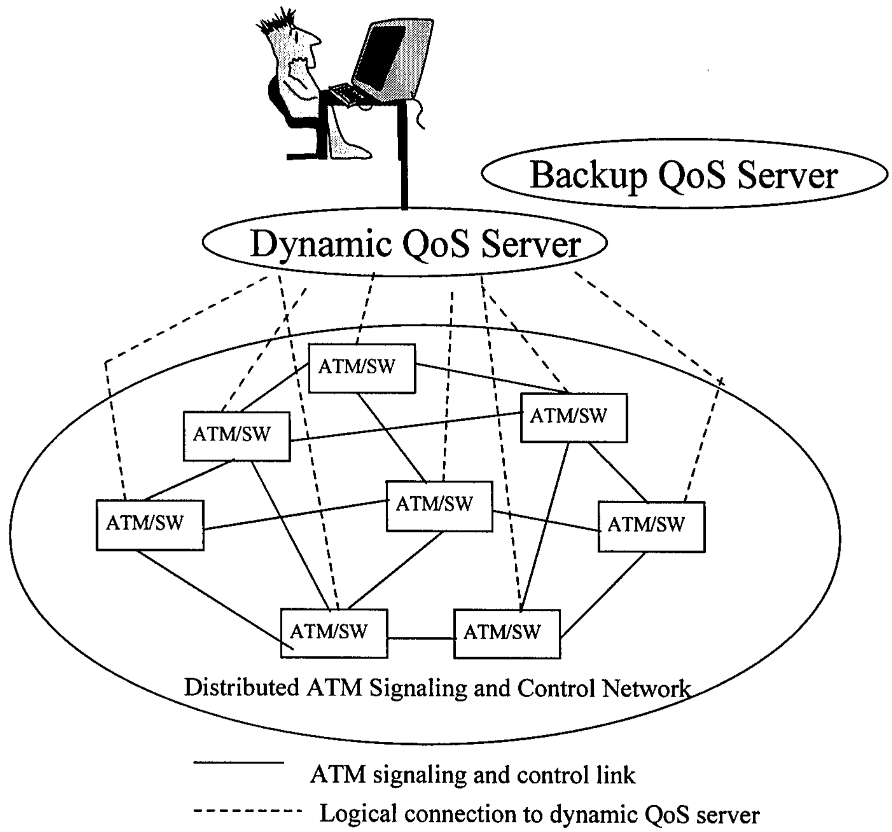

Figure 1. Distributed Command and Control System with Dynamic QoS Server System.

3. Control Algorithms for Dynamic Bandwidth Management and Service Restoration (Phase I Work)

3.1. Dynamic Bandwidth Management

When an existing connection requests for bandwidth increment, the QoS server will first look up the existing routing path to see if all ATM switches along this path have sufficient capacity to accommodate the request. If it does, the QoS server will send resource adjustment message to two end ATM switches of the routing path to make bandwidth adjustment. Intermediate ATM switches along the path do not involve the bandwidth adjustment decision process. The source ATM switch then instructs the users to start sending the requested increased traffic. If any link along the path can't accommodate the request, the request will be rejected. Some algorithms proposed by Ref. [3] are used as the reference algorithms for this study.

3.2 Self-Healing Control

The command and control system should operate normally even when the network components fail. In this proposal, we use the PNNI signaling capability to help implement the self-healing capability. As mentioned earlier, the current version of PNNI

[2] is capable of re-routing new call requests, not existing connections. However, its crank-back signaling capability can be used here as a failure detection mechanism of the proposed self-healing scheme. The proposed self-healing network control process is described as follows. If a network component fails, an PNNI's crank-back signaling message is sent back to the source ATM switch of the connection, which would trigger the restoration message sent from the source ATM switch to the QoS server. The QoS server will then compute an alternate path with the "best available QoS" if the connection is a high-priority connection, and it sends the Restore-ACK message back to the source ATM switch to trigger the ATM traffic re-routing. If the QoS associated with high-priority connections being restored is not the original QoS, the network will try to restore its requested QoS level as quickly as the network allows or by the operator intervention. For the low-priority connection, if the network spare capacity can't accommodate its requested QoS level, the low-priority connection will not be restored. Several algorithms and procedures that could be available for this service restoration implementation can be found in Ref. [4-7]. Note that, although many ATM networks have been deployed in USA, there is no ATM restoration system available at the moment this proposal is written.

3.3 Core Features for Phase I Feasibility Study

The following core capabilities will be studied and demonstrated through a testbed to show the feasibility of implementing the core function of ATM-based adaptive command and control system.

Bandwidth-on-Demand

- Traffic monitoring for ATM Virtual Paths (VPs);
- Increase bandwidth if the requested is made and the available bandwidth can meet its QoS; and
- Remotely send commands to users for acknowledgment of the receipt of bandwidth adjustment request.

Service Restoration

In addition to above three features, service restoration requires the network to have the capability to update ATM switches' VPI/VCI tables when needed (for traffic re-routing).

4. Modeling and Testing Environment

Figure 2 depicts a modeling and testing environment for real-time traffic monitoring and bandwidth allocation. Two Fore System's ATM switches with LANE server capabilities will provide non-blocking, guaranteed bandwidth among workstations. An HP Broadband ATM tester is used to generate test data to saturate the test link and to monitor traffic conditions. An HP Openview SNMP manager will be used in real-time to collect traffic parameters from the Fore switches and Sun Workstations. All information

collected will be stored in HP's database and these database information will be queried by the decision making PC for further processing.

An IBM PC is used to implement the real-time control module and state-tables. This PC will request its interested SNMP information from HP Openview workstation and stored them in its local SQL database. The control module will use the collected SNMP information from the SQL database to update the state-tables which are also stored in the SQL database. The state-tables will then be used by the control module along with the "scenarios", set points, threshold, etc., we defined to determine the actions to be sent to the HP Openview workstation to correct the current network conditions or pro-act to future conditions.

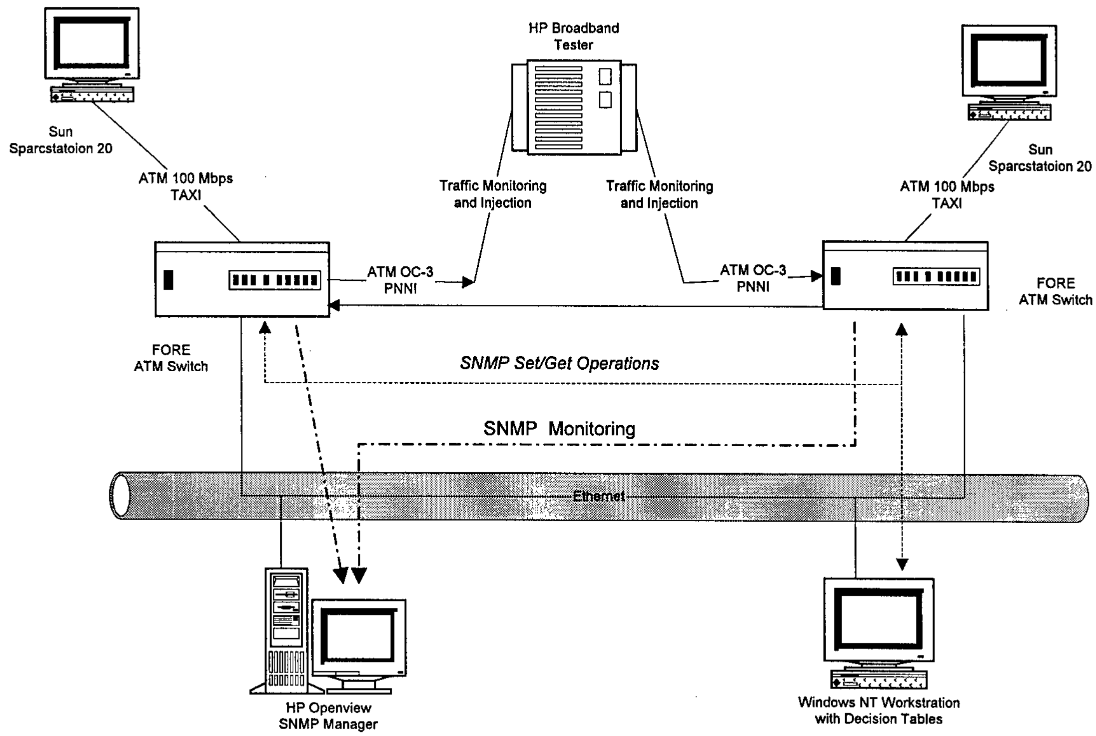

Figure 2. Real-Time Traffic Monitoring and Bandwidth Management Testing Configuration.

5. Feasibility Demonstration and Procedures

5.1 Description of Demonstrated Systems

For the testbed, we first setup ATM evaluation environment as described in Section 4 (also in October's Interim Status Report [8]) using Fore System's ASX-200 ATM Switches. Following the initial system setup, we investigated the database structure of HP Openview, remedy on the HP SNMP manager workstation, virtual LAN functionality to

support logical separation and group association in the legacy LAN environment and some HP Openview Application Programming Interfaces (APIs) based on the HP J210 platform. The Carnegie Mellon University's (CMU's) SNMP toolkit, which allows direct SNMP access operations, Set, Get, etc., via ethernet network to the vendor equipment, is selected for our testbed due to its features and flexibility. We also installed and setup X-Window emulation package, X-Vision, on a PC to perform remote connectivity to HP J210 for program development. Some enhancements have been added to the CMU's SNMP's toolkit to allow us to remotely access, setup, and retrieve ATM VP/VC and other quality of Service parameters directly to and from the Fore's ASX-200 ATM Switch.

5.2 Procedures for Feasibility Study

This section describes Procedures for Building Direct SNMP Control of Fore Systems ASX-200 ATM Switch. These procedures list the steps to use Carnegie Mellon University's (CMU) SNMP tools to build an "executable" to remotely "set" and "get" SNMP parameters. This "executable" is further verified by accessing the Fore Systems ASX-200 Asynchronous Transfer Mode (ATM) switches to setup the Virtual Path (VP) and Virtual Channels (VC) on the individual interfaces.

The computer platform that we based our research upon is:

- Hewlett Packard J210 9000/770 with 256 Mbytes DRAM,
- HPUX Version 10.0,
- HP Openview 4.11, and
- Fore ASX-200 SNMP MIB Ver. 1.40.

I. **Files:**

```
1)  In directory /home/yiq/v1/apps:
    Makefile snmpget.c snmpwalk.c snmpset.c
    mib file: rfc1213-MIB-II(mib-2 file),
              fore-common.mib,
              fore-switch-notrap.mib, (mib file for fore-atm-switch),
              rfc1696.mib, and
              ascend.mib. (mib file for ASCEND MAX)
    The fore-switch-notrap.mib is modified from fore-switch.mib. We just delete the
    trap part of the fore-switch.mib since the parse.c will not recognize the syntax
    of trap definition and we do not use the trap now.

2)  In directory /home/yiq/v1/snmplib
    Makefile, acl.c, context_parse.c,  party.c, snmp_auth.c, acl_parse.c, md5.c,
    party_parse.c, snmp_client.c, asn1.c, mib.c, snmp.c, system.c, context.c, parse.c,
    snmp_api.c, view.c, acl.h, md5.h, party.h, snmp_client.h, view.h, asn1.h, mib.h,
    snmp.h, snmp_impl.h, context.h, parse.h, snmp_api.h, system.h
```

II. **To compile**

```
1)  dir /home/yiq/v1/snmplib    make

2)  dir /home/yiq/v1/apps    make
```

III. **To load more mib file**

dir /home/yiq/v1/snmplib and vi parse.c, in load_mib() function, add more strcpy(file_list[i], mib-file-name) into the load_mib() the function.

Those file_list should be listed in a particular order. i.e. fore-switch-notrap.mib uses defintion of fore-common.mib, so the fore-common.mib should be listed before fore-switch-notrap.mib. Therefore, we first parse the fore-common.mib and add the defintions into a table, when we parse the fore-switch-notrap.mib, we can search the information in the table to recognize the object.

IV. To Run dir /home/yiq/v1/apps

```
1)  To snmpget, type
    ./snmpget -v 1 hostname community objectID [objectID]; i.e. ./snmpget -v 1
    nms1.linkmax.com public sysName.0 sysContact.0; or
    ./snmpget -v 1 206.197.95.100 fun sysName.0 (The community name for reading
    206.197.95.100 is fun)

2)  To snmpwalk, type
     ./snmpwalk -v 1 hostname community [objectID], i.e.
     ./snmpwalk -v 1 nms1.linkmax.com public system; or
     ./snmpwalk -v 1 206.197.95.100 fun system

3)  To snmpset, type
    ./snmpset -v 1 hostname community [objectID type value]+
    where type is one of: i, s, x, d, n, o, t, a
        i: INTEGER,
        s: STRING,
        x: HEX STRING,
        d: DECIMAL STRING,
        n: NULLOBJ,
        o: OBJID,
        t: TIMETICKS,
        a: IPADDRESS
    i.e. ./snmpset -v 1 206.197.95.100 fun sysName.0 s "asx-200"
```

We can use snmpset to update a table of ATM switch, i.e. to delete or add a row ATM MIB Types Definitions -- The status of a table entry. -- Taken from RFC1271-MIB definitions:

```
EntryStatus ::= INTEGER     { valid(1),      createRequest(2),       underCreation(3),
invalid(4)      }
```

Setting this object to the value invalid(4) has the effect of invalidating the corresponding entry. That is, it effectively disassociates the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries currently not in use. Proper interpretation of such entries requires examination of the relevant EntryStatus object.

An existing instance of this object cannot be set to createRequest(2). This object may only be set to createRequest(2) when this instance is created. When this object is created, the agent may wish to create supplemental object instances to complete a conceptual row in this table. Immediately after completing the create operation, the agent must set this object to underCreation(3). Entries shall exist in the underCreation(3) state until the management station is finished configuring the entry and sets this object to valid(1) or aborts, setting this object to invalid(4). If the agent determines that an entry has been in the underCreation(3) state for an abnormally long time, it may decide that the management station has crashed. If the agent makes this decision, it may set this object to invalid(4) to reclaim the entry. A prudent agent will understand that the management

station may need to wait for human input and will allow for that possibility in its --determination of this abnormally long period.

For example, the virtual path table is defined in fore-switch.mib.

```
pathRouteTable OBJECT-TYPE  SYNTAX  SEQUENCE OF PathRouteEntry ACCESS  not-accessible
    STATUS  mandatory  DESCRIPTION "A table of information about the routing of paths
    through  this ATM switch."  ::= { pathGroup 2 }
pathRouteEntry OBJECT-TYPE  SYNTAX  PathRouteEntry ACCESS  not-accessible
    STATUS  mandatory  DESCRIPTION  "A table entry containing path route information."
    INDEX   { pathrInputPort, pathrInputVPI, pathrOutputPort, pathrOutputVPI  }
    ::= { pathRouteTable 1 }
PathRouteEntry ::=  SEQUENCE {
    pathrInputPort          INTEGER,
    pathrInputVPI           INTEGER,
    pathrOutputPort         INTEGER,
    pathrOutputVPI          INTEGER,
    pathrStatus             EntryStatus,
    pathrMaxBandwidth       INTEGER,
    pathrAllocBandwidth     Gauge,
    pathrCells              Counter,
    pathrUptime             TimeTicks,
    pathrSigProtocol        AtmSigProtocol,
    pathrRejectedCells      Counter,
    pathrTrafficShapeVPI    INTEGER,
    pathrUpcContract        INTEGER        }
```

```
a)  After telnet to the 206.197.95.100
    We can show the virtual path on port 1C1 by:
    localhost::configuration vpc> show 1c1
        Input      Output
        Port  VPI  Port VPI    MaxBW     BW MaxVCs  VCs UPC Shape Prot  VBROB BuffOB
        originate
        1C1    0               45.0M   0.0K    256   0  N/A       pvc    100    100
        1C1    1    1C2   2     0.0K   0.0K    N/A  N/A   0       pvc     N/A   N/A
        1C1    4    1C2   5     0.0K   0.0K    N/A  N/A   1     0 pvc     N/A   N/A

b)  To delete an entry
    ./snmpset -v 1 206.197.95.100 fun pathrStatus.16.4.17.5 i 4
        pathrStatus is an attribute of PathRouteEntry with EntryStatus syntax. We snmpset
        the pathrStatus object id with index 16.4.17.5 (which means pathrInputPort
        16(1C1), pathrInputVPI 4, pathrOutputPort 17(1C2), pathrOutputVPI 5) to
        4(invalid).

    To verify the result:
    localhost::configuration vpc> show 1c1
        Input      Output
        Port  VPI  Port VPI    MaxBW     BW MaxVCs  VCs UPC Shape Prot  VBROB BuffOB
        originate
        1C1    0               45.0M   0.0K    256   0  N/A       pvc    100    100
        1C1    1    1C2   2     0.0K   0.0K    N/A  N/A   0       pvc     N/A   N/A
    A row is deleted.

c)  To add an entry
    ./snmpset -v 1 206.197.95.100 fun pathrStatus.16.4.17.5 i 2
    to set the pathrStatus object id with index 16.4.17.5 to createRequest(2)

    To verify the result:
    localhost::configuration vpc> show 1c1
        Input      Output
        Port  VPI  Port VPI    MaxBW     BW MaxVCs  VCs UPC Shape Prot  VBROB BuffOB
        originate
        1C1    0               45.0M   0.0K    256   0  N/A       pvc    100    100
        1C1    1    1C2   2     0.0K   0.0K    N/A  N/A   0       pvc     N/A   N/A
       *1C1    4    1C2   5     0.0K   0.0K    N/A  N/A   0     0 pvc     N/A   N/A
    A new marked virtual path(1c1 4 1c2 5) is added into the table. Immediately after
    completing the create operation, the switch sets this object to underCreation(3).
    We type ./snmpget -v 1 206.197.95.100 fun pathrStatus.16.4.17.5 and can see
```

```
enterprises.fore.systems.atmSwitch.software.asxd.pathGroup.pathRouteTable.pathRo
uteEntry.pathrStatus.16.4.17.5 = underCreation(3)
```

d) To make the entry valid:

```
./snmpset -v 1 206.197.95.100 fun pathrStatus.16.4.17.5 i 4
To set the pathrStatus object id with index 16.4.17.5 to valid(1).

To verify the result:
localhost::configuration vpc> show 1c1
    Input       Output
    Port  VPI  Port VPI    MaxBW      BW MaxVCs   VCs UPC Shape Prot   VBROB BuffOB
    originate
    1C1    0               45.0M    0.0K    256    0  N/A        pvc    100     100
    1C1    1    1C2   2     0.0K    0.0K    N/A  N/A   0         pvc     N/A    N/A
    1C1    4    1C2   5     0.0K    0.0K    N/A  N/A   0      0  pvc     N/A    N/A
Now the path 1c1 4 1c2 5 is valid.
```

6. Significance to Phase II Program

The testbed developed by MMCI in the Phase I work allows us to demonstrate selected key features of dynamic bandwidth management (through a table decision process) and network restoration (through the update of VPI/VCI tables). These demonstrated features are core of advanced network control capabilities proposed in Phase II work, including. adaptive QoS adjustment, dynamic bandwidth management, priority call admission control, automatic service restoration and Web-based operator supporting system.

7. Summary

We have reviewed the work that has been conducted under Phase-I contract, which includes the architecture concept of an adaptive QoS-based ATM distributed command and control system architecture which would support Navy's mission critical operations whenever and wherever needed. This architecture is based on the novel application of proven real-time control methodology, applied by DARPA, Navy and private sectors on projects as diverse as high-speed signaling transport, adaptive QoS control and management, and Web-based signaling and control. We also describe a testbed used to demonstrate selected key features of dynamic bandwidth management and network restoration. The completion of Phase I work help facilitate the progress and increase the probability of success of Phase II work which designs and prototypes an advanced distributed command and control system architecture capable of providing priority call admission control and adaptive QoS control for mission critical applications whenever and wherever needed, and automatic service restoration capability to keep the crucial command and control system intact when the network component fails. The results and prototype system developed from the Phase II work, if completed, would help Navy provide the capability of supporting mission critical applications whenever and wherever needed under both the normal and network failure conditions. They can also be used in private sectors to help service providers provide cost-effective private line services, survivable premium services and QoS-based premium wireless services.

8. References

1. The ATM Forum Technical Committee, Private Network-Network Interface Specification Version 1.0, (PNNI 1.0), af-pnni-0055.000, March 1996.
2. Wu, T-H., Kopeglu, I., and Chang, B., “Distributed Interactive Video System Designs and Analysis,” *IEEE Communications Magazine*, March 1997.
3. Kawamura, R., et., “Fast VP-Bandwidth Management with Distributed Control in ATM Networks,” IEICE Trans. Communications, Vol. E77-B, January 1994.
4. Wu, T-H., and Yoshikai, N., *ATM Transport and Network Integrity*, Academic Press Inc., June 1997.
5. Wu, T-H., *Fiber Network Service Survivability*, ArchTech House, 1992.
6. Wu, T-H., "Emerging Technologies for Fiber Network Survivability," *IEEE Communications Magazine*, February 1995.
7. Hsing, D., et al, “A Restoration System based on Pre-planned Source Routing in ATM Networks,” Proceedings of ICC, June 1997.
8. Interim Status report on SBIR No. N0024-96-C-4188, October 1996.